

My url repo in DevAzure :

https://dev.azure.com/VolodymyrDibrova0326/MyProjectVivi/_git/fullstack-azure-app

Practical Task 1: Create and Initialize a New Repository with a README

Practical Task 2: Implement Branching and Merging

The screenshot shows two main sections of the Azure DevOps interface. The top section displays a repository named 'MyEnhancedRepo' under 'MyProjectVivi'. It shows two files: 'FEATURES.md' and 'README.md'. The bottom section shows a pull request titled 'Added FEATURES.md' from 'feature-enhancement' into the 'main' branch. The pull request has been completed by Volodymyr Dibrova. The merge commit message is 'Merged PR 5: Added FEATURES.md'. The pull request summary indicates 'No merge conflicts' and 'Last checked Just now'. The pull request details also show the description 'Added FEATURES.md'.

Practical Task 3: Set Up Branch Policies and Code Review

Requirements:

- Configure branch policies for the **main** branch in Azure Repos to require at least one reviewer for pull requests and to enforce a minimum number of successful builds (you can simulate this by using a mock build status if needed).
- Create a new branch named **bugfix-issue** from the **main** branch and make changes to the **FEATURES.md** file to address a hypothetical bug (e.g., update a feature description).
- Commit the changes and push the **bugfix-issue** branch to the remote repository.
- Create a pull request from **bugfix-issue** to **main**, ensuring that the pull request meets the branch policies set earlier.
- Invite at least one team member or colleague to review the pull request and provide feedback.
- Address any comments or requested changes from the reviewer and update the pull request accordingly.

- Merge the pull request into the **main** branch once it has been approved.

1. Configured and setted a 2 number of reviewers

The screenshot shows the GitHub Project Settings interface for the repository **MyEnhancedRepo**. The left sidebar lists project settings categories like General, Boards, and Pipelines. The main area displays repository-specific settings. Under the **Policies** tab, there are three sections: **Limit merge types** (disabled), **Build Validation** (enabled for the **MyProjectVivi** trigger), and **Status Checks** (disabled). A note at the top right says: "Check to see that all comments have been resolved on pull requests."

This screenshot shows the branch policies for the **main** branch of **MyEnhancedRepo**. It highlights the **Require a minimum number of reviewers** policy, which is set to **On** with a value of **2**. Other options include checkboxes for allowing self-approval, prohibiting self-approval, and allowing completion even if some reviewers vote to wait or reject.

The screenshot shows a pull request titled **Updated FEATURES.md** from **bugfix-issue** into **main**. The pull request has 17 approvals. On the right, the **Reviewers** section indicates that no required reviewers have been added. The **Tags** section shows that no tags are present.

Name	Date modified	Type	Size
FEATURES	2/18/2025 5:47 PM	Markdown Source ...	0 KB
README	2/18/2025 12:28 PM	Markdown Source ...	1 KB

```

PS E:\projects\terraformProj\DevopsService\task1\MyEnhancedRepo> cd ../
PS E:\projects\terraformProj\DevopsService\task1\MyEnhancedRepo> git pull
remote: Azure Repos
remote: We noticed you're using an older version of Git. For the best experience, upgrade to a newer version.
remote: Found 7 objects to send. (1 ms)
Unpacking objects: 100% (7/7), 839 bytes | 25.00 KiB/s, done.
From https://dev.azure.com/VolodymyrDibrova0326/MyProjectVivi/_git/MyEnhancedRepo
  71136d5..8cfee9f  main      -> origin/main
 * [new branch]    bugfix-issue -> origin/bugfix-issue
Updating 71136d5..8cfee9f
Fast-forward
 FEATURES.md | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 FEATURES.md
PS E:\projects\terraformProj\DevopsService\task1\MyEnhancedRepo>

```

Practical Task 4: Create a Simple Build Pipeline in Azure DevOps Requirements:

1. Log in to Azure DevOps:

- a. Access your Azure DevOps organization and navigate to your project.

2. Create a Git Repository:

- a. In Azure Repos, create a new Git repository named MySampleApp.
- b. Clone the repository to your local machine and add a simple application (e.g., a basic .NET Core or Node.js application).

3. Create a Build Pipeline:

- a. Navigate to Azure Pipelines and create a new pipeline.
- b. Select the Azure Repos Git repository as the source.c. Use the YAML pipeline editor to define the pipeline, including the following steps:
dotnet restore (for .NET) or npm install (for Node.js) to restore dependencies.
dotnet build (for .NET) or npm run build (for Node.js) to build the application.
dotnet test (for .NET) or a testing command for Node.js to run unit tests.

4. Run the Pipeline:

- a. Save and run the pipeline.
- b. Verify that the build pipeline completes successfully and check the build logs for

any
error

rova0326 / MyProjectVivi / Repos / Files / MySampleApp

```

MySampleApp
  azure-pipelines.yml
    index.js
    package.json
  README.md

trigger:
- main

pool:
  name: MyLinux
  demands:
    - agent.name -equals myAgent

tasks:
- task: NodeTool@0
  inputs:
    versionSpec: '16.x'
  displayName: 'Install Node.js'

- script: npm install
  displayName: 'Install dependencies'

- script: npm run build
  displayName: 'Build the application'

- script: npm test
  displayName: 'Run tests'

- task: PublishBuildArtifacts@1
  displayName: "Publish Build Artifacts"
  inputs:
    pathToPublish: 'dist'
    artifactName: 'drop'
    publishLocation: 'container'

```

Contents History Compare Blame

File Explorer

Name	Date modified	Type	Size
package	2/18/2025 6:03 PM	JSON Source File	1 KB
README	2/18/2025 6:01 PM	Markdown Source ...	1 KB
index	2/18/2025 6:03 PM	JSON Source File	1 KB

Windows PowerShell

```

create mode 100644 index.json
create mode 100644 package.json
PS E:\projects\terraformProj\DevopsService\task1\MySampleApp> git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 487 bytes | 487.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Analyzing objects... (4/4) (4 ms)
remote: Validating commits... (1/1) done (0 ms)
remote: Storing packfile... done (40 ms)
remote: Storing index... done (35 ms)
remote: We noticed you're using an older version of Git. For the best experience, upgrade to a newer
version.
To https://dev.azure.com/VolodymyrDibrova0326/MyProjectVivi/_git/MySampleApp
  81a3a67..d8cbf4d  main -> main
PS E:\projects\terraformProj\DevopsService\task1\MySampleApp> |

```

Azure DevOps VolodymyrDibrova0326 / MyProjectVivi / Repos / Files / MySampleApp

MyProjectVivi

- Overview
- Boards
- Repos**
 - Files
 - Commits
 - Pushes
 - Branches
 - Tags
 - Pull requests
 - Advanced Security

MySampleApp

- index.json
- package.json
- README.md

main / Type to find a file or folder...

Files

Contents History

Name	Last change	Commits
index.json	Just now	d8cbf4d2 first commit for sample app ...
package.json	Just now	d8cbf4d2 first commit for sample app ...
README.md	19m ago	81a3a67b Added README.md Volody...

Introduction
TODO: Give a short introduction of your project. Let this section explain the objectives or the motivation behind this project.

Getting Started

Azure DevOps VolodymyrDibrova0326 / MyProjectVivi / Pipelines / MySampleApp / 20250218.1

MyProjectVivi

- Overview
- Boards
- Repos**
- Pipelines**
- Pipelines
- Environments
- Library
- Test Plans
- Artifacts

Jobs in run #20250218.1
MySampleApp

Job

1 Pool: myLinux
2 Agent: myAgent
3 Started: Just now
4 Duration: 25s
5
6 Job preparation parameters

Initialize job 4s
Checkout MySampleApp... 4s
Install Node.js 7s
Install dependencies 2s
Build the application 2s
Run tests 2s
Post-job: Checkout M... <1s

View raw log

← → ⌂ dev.azure.com/VolodymyrDibrova0326/b9951ba4-faf2-40e6-bfb0-c8b66bfd55e/_apis/build/builds/145/logs/13

Как думать по SQL... Webdesign - Googl... Настройки Spring Boot: Excel...

```
2025-02-18T16:33:57.029218Z > INFO Running tests...
2025-02-18T16:33:57.0292453Z
2025-02-18T16:33:57.0365102Z Running tests...
2025-02-18T16:33:57.0538442Z ##[section]finishing: Run tests
2025-02-18T16:33:57.0556719Z ##[section]starting: Run the application
2025-02-18T16:33:57.0562166Z =====
2025-02-18T16:33:57.0562226Z Task : Command Line
2025-02-18T16:33:57.0562277Z Description : Run command line script using Bash on Linux and macOS and cmd.exe on Windows
2025-02-18T16:33:57.0562473Z Version : 2.250.1
2025-02-18T16:33:57.0562551Z Author : Microsoft Corporation
2025-02-18T16:33:57.0562634Z Help : https://docs.microsoft.com/azure/devops/pipelines/tasks/utility/command-line
2025-02-18T16:33:57.0562744Z =====
2025-02-18T16:33:57.0656104Z ##[command]/home/azureuser/myagent/externals/node20_1/bin/node -v
2025-02-18T16:33:57.0656342Z /home/azureuser/myagent/externals/node20_1/bin/node: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.28' not found (required by /home/azureuser/myagent/externals/node20_1/bin/node)
2025-02-18T16:33:57.0957744Z ##[warning]The agent operating system doesn't support Node20. Using Node16 instead. Please upgrade the operating system of the agent to remain compatible with the latest Node.js releases.
2025-02-18T16:33:59.0175939Z Generating script.
2025-02-18T16:33:59.0175967Z Script contents:
2025-02-18T16:33:59.0176007Z npm start
2025-02-18T16:33:59.0176493Z =====
2025-02-18T16:33:59.01785687Z [command]/bin/bash -nopro
2025-02-18T16:33:59.0180221Z
2025-02-18T16:33:59.0180892Z > mysampleapp1.0.0 start
2025-02-18T16:33:59.0180892Z > node index.js
2025-02-18T16:33:59.0182119Z
2025-02-18T16:33:59.05624105Z Hello from MySampleApp!
2025-02-18T16:33:59.0582169Z ##[section]finishing: Run
2025-02-18T16:33:59.0584145Z ##[section]starting: Check
2025-02-18T16:33:59.05844764Z =====
2025-02-18T16:33:59.05844907Z Task : Get sources
2025-02-18T16:33:59.05844907Z Description : Get sources
2025-02-18T16:33:59.05845177Z Version : 1.0.0
2025-02-18T16:33:59.05845167Z Author : Microsoft
2025-02-18T16:33:59.05845236Z Help : [More Information]
2025-02-18T16:33:59.05845343Z =====
2025-02-18T16:33:59.08117318Z Cleaning any cached credentials
2025-02-18T16:33:59.08220099Z ##[section]finishing: Clean
2025-02-18T16:33:59.08242868Z ##[section]starting: Final
2025-02-18T16:33:59.08251786Z Cleaning up task key
2025-02-18T16:33:59.08252794Z Start cleaning up orphan processes
2025-02-18T16:33:59.08455035Z ##[section]finishing: Final
2025-02-18T16:33:59.08478105Z ##[section]finishing: Job
```

Practical Task 5: Set Up Continuous Deployment (CD) to Azure Web App

Requirements:

1. Extend the Build Pipeline:

- Open the existing build pipeline created in Task 4.
- Add a new stage for deployment after the build stage.

2. Create an Azure Web App:

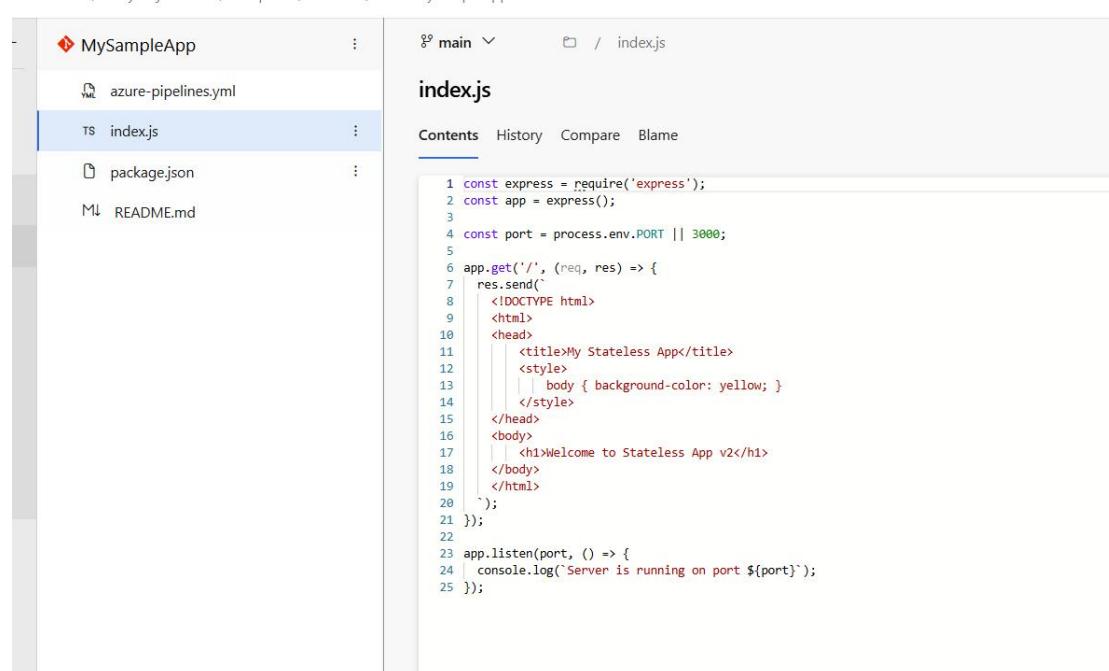
- In the Azure portal, manually create an Azure Web App or use an ARM template to create it.

3. Configure Release Pipeline:

- Create a release pipeline in Azure DevOps that triggers automatically after a successful build.
- Add an Azure Web App deployment task to the release pipeline, specifying the Azure subscription and the Web App name.

4. Deploy the Application:

- Save and run the release pipeline.
- Verify that the application is successfully deployed and accessible via the web



The screenshot shows the Azure DevOps code editor interface. On the left, there's a file tree for a project named "MySampleApp". The files listed are "azure-pipelines.yml", "index.js", "package.json", and "README.md". The "index.js" file is currently selected and its content is displayed on the right. The code in "index.js" is as follows:

```
1 const express = require('express');
2 const app = express();
3
4 const port = process.env.PORT || 3000;
5
6 app.get('/', (req, res) => {
7   res.send(`
8     <!DOCTYPE html>
9     <html>
10       <head>
11         <title>My Stateless App</title>
12         <style>
13           body { background-color: yellow; }
14         </style>
15       </head>
16       <body>
17         <h1>Welcome to Stateless App v2</h1>
18       </body>
19     </html>
20   `);
21 });
22
23 app.listen(port, () => {
24   console.log(`Server is running on port ${port}`);
25 });
```

Azure DevOps Pipeline Overview:

- Release**: Manually triggered by Volodymyr Dibrova on 18.02.2025, 21:12.
- Stages**: Stage 5 (Succeeded) is highlighted.
- Artifacts**: MySampleApp (47) - 20250218.1.
- Deployment Details**:
 - Deployment succeeded on 18.02.2025, 21:14.
 - Run on agent - Succeeded: 4/4 task(s) succeeded.
 - 2 warnings:
 - The agent operating system doesn't support Node20. Using Node16 instead. Please upgrade th...
 - The agent operating system doesn't support Node20. Using Node16 instead. Please upgrade th...
- Associated changes**: View commits and work items for _MySampleApp (47) / 20250218.1.

```

1 2025-02-18T20:03:35.547350Z ##[section]Starting: Deploy Azure App Service
2 2025-02-18T20:03:35.549649Z -----
3 2025-02-18T20:03:35.549695Z Task : Azure App Service deploy
4 2025-02-18T20:03:35.549724Z Description : Deploy to Azure App Service a web, mobile, or API app using Docker, Java, .NET, .NET Core, Node.js, PHP, Python, or F
5 2025-02-18T20:03:35.549742Z Version : 4.247.1
6 2025-02-18T20:03:35.549800Z Author : Microsoft Corporation
7 2025-02-18T20:03:35.549827Z Help : https://aka.ms/azureappservicetroubleshooting
8 2025-02-18T20:03:35.549857Z -----
9 2025-02-18T20:03:35.589574Z ##[command]/home/azureuser/magent/externals/node20_1/bin/node -v
10 2025-02-18T20:03:35.589574Z /home/azureuser/magent/externals/node20_1/bin/node: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.28' not found (required by /)
11 2025-02-18T20:03:36.396942Z ##[warning]The agent operating system doesn't support Node20. Using Node16 instead. Please upgrade the operating system of the agent
12 2025-02-18T20:03:40.377839Z Go service connection details for Azure App Service: 'my-sample-app-123'.
13 2025-02-18T20:03:50.094605Z Package deployment using ZIP Deploy initiated.
14 2025-02-18T20:04:03.747962Z Deploy logs can be viewed at https://my-sample-app-123-azcud7bcdtfye2hu.scm.westeurope-01.azurewebsites.net/api/deployments/22d12cd...
15 2025-02-18T20:04:03.748152Z Successfully deployed web package to App Service.
16 2025-02-18T20:04:21.179651Z Successfully updated App Service configuration details
17 2025-02-18T20:04:24.329620Z Successfully updated deployment History at https://my-sample-app-123-azcud7bcdtfye2hu.scm.westeurope-01.azurewebsites.net/api/deploy...
18 2025-02-18T20:04:24.688639Z App Service Application URL: https://my-sample-app-123-azcud7bcdtfye2hu.westeurope-01.azurewebsites.net
19 2025-02-18T20:04:26.950910Z ##[section]Finishing: Deploy Azure App Service
20

```

my-sample-app-123 | Deployment Center

Deployment Center Settings

- Source: Azure Repos (MyProjectVivi)
- Build provider: Azure Pipelines

Deployment Center Settings

- Source: Azure Repos (MySampleApp)
- Branch: refs/heads/main
- Build provider: Azure Pipelines

Practical Task 6: Implement CI/CD with GitHub and Azure Pipelines

Requirements:

1. Create a GitHub Repository:

- Navigate to GitHub and create a new repository named MySampleApp.

2. Push Application Code:

- Push the application code from Task 4 to the new GitHub repository.

3. Set Up Azure Pipeline:

- In Azure DevOps, create a new pipeline that uses the GitHub repository as the

source.b. Authenticate with GitHub and select the repository.

4. Define the Pipeline:

- Use the YAML pipeline editor to define the pipeline with steps to:

i. Build the application.

ii. Run tests.

iii. Deploy to the Azure Web App (similar to Task 5).

5. Enable GitHub Integration:

- Configure the pipeline to trigger automatically on every push to the main branch.

- Verify that the pipeline triggers successfully and the application is deployed to Azure.

Created a GitHub and pushed from Dev Azure to GitHub

```
PS E:\projects\terraformProj\DevopsService\taskService\MySampleApp> git remote -v
origin https://VolodymyrDibrova0326@dev.azure.com/VolodymyrDibrova0326/MyProjectVivi/.git/MySampleApp (fetch)
origin https://VolodymyrDibrova0326@dev.azure.com/VolodymyrDibrova0326/MyProjectVivi/.git/MySampleApp (push)
PS E:\projects\terraformProj\DevopsService\taskService\MySampleApp> git remote remove origin
PS E:\projects\terraformProj\DevopsService\taskService\MySampleApp> git remote add origin https://github.com/Vivien87/MySampleApp.git
PS E:\projects\terraformProj\DevopsService\taskService\MySampleApp> git remote -v
origin https://github.com/Vivien87/MySampleApp.git (fetch)
origin https://github.com/Vivien87/MySampleApp.git (push)
PS E:\projects\terraformProj\DevopsService\taskService\MySampleApp> git push -u origin main
To https://github.com/Vivien87/MySampleApp.git
 ! [rejected]      main > main (fetch first)
error: failed to push some refs to 'https://github.com/Vivien87/MySampleApp.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
PS E:\projects\terraformProj\DevopsService\taskService\MySampleApp> git push -u origin main --force
Enumerating objects: 50, done.
Counting objects: 100% (50/50), done.
Delta compression using up to 8 threads
Compressing objects: 100% (48/48), done.
Writing objects: 100% (50/50), 5.37 KiB | 917.00 KiB/s, done.
Total 50 (delta 30), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (30/30), done.
To https://github.com/Vivien87/MySampleApp.git
```

github.com/vivien87/MySampleApp

Как думать на SQL?... Webdesign - Googl... Настройки Spring Boot: Excel,...

Vivien87 / MySampleApp

Type to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

MySampleApp Public

Pin Unwatch 1

main 1 Branch 0 Tags Go to file Add file Code

Volodymyr Dibrova and Volodymyr Dibrova Updated azure-pipelines.yml 26bd9a9 · 1 hour ago 17 Commits

README.md Added README.md yesterday

azure-pipelines.yml Updated azure-pipelines.yml 1 hour ago

index.js Updated index.js yesterday

package.json Updated package.json yesterday

README

Introduction

TODO: Give a short introduction of your project. Let this section explain the objectives or the motivation behind this project.

github.com/Vivien87/MySampleApp/blob/main/azure-pipelines.yml

Как думать на SQL?... Webdesign - Googl... Настройки Spring Boot: Excel,...

Files

main Go to file

README.md

azure-pipelines.yml

index.js

package.json

MySampleApp / azure-pipelines.yml

Vivien87 Update azure-pipelines.yml

Code Blame 34 lines (26 loc) · 653 Bytes Code 55% faster with GitHub Copilot

```
trigger:
  - main

pool:
  name: MyLinux
  demands:
    - agent.name -equals myAgent

steps:
  - checkout: self

  - script: ls -la $(Build.SourcesDirectory)
    displayName: 'Checking file in repos'

  - task: NodeTool@0
    inputs:
      versionSpec: '16.x'
    displayName: 'Install Node.js'

  - script: npm install
    displayName: 'Install dependencies'

  - script: npm run build
    displayName: 'Build the application'
```

Azure DevOps Pipelines summary for Vivien87.MySampleApp (59) run 20250219.1. The run was manually triggered by Volodymyr Dibrova. It shows a green checkmark for the task and a warning message: "This run is being retained as one of 3 recent runs by pipeline." The pipeline details include the repository (Vivien87/MySampleApp), commit (main, 3cdcd073), and duration (Just now, 37s). A section for "Warnings" lists four items related to Node.js compatibility.

Azure DevOps Releases screen showing a new release pipeline (2) for Release-1. The pipeline status is "Succeeded". The "Release" section shows it was triggered manually by Volodymyr Dibrova on 19.02.2025, 23:53. The "Artifacts" section shows an artifact named "Vivien87.MySampleApp..." from the main branch. The "Stages" section shows Stage 2 has succeeded with 2 warnings on the same date.

Azure DevOps Pipelines log for the "Azure App Service Deploy: my-sample-app-123" task. The log shows the deployment process starting at 2025-02-19T21:54:35. The log output includes several warning messages about Node.js compatibility and service connection details. The deployment was successful, as indicated by the green checkmark icon.

```

2025-02-19T21:54:35.456Z [##[section]Starting: Azure App Service Deploy: my-sample-app-123
2025-02-19T21:54:35.4889869Z
2025-02-19T21:54:35.4889869Z Task : Azure App Service Deploy
2025-02-19T21:54:35.4889869Z Description : Deploy to Azure App Service a web, mobile, or API app using Docker, Java, .NET Core, Node.js, PHP, Python, or Ruby
2025-02-19T21:54:35.4889869Z Version : 2.222.5
2025-02-19T21:54:35.4889869Z Author : Microsoft Corporation
2025-02-19T21:54:35.4889869Z Help : https://aka.ms/azureappservice/troubleshooting
2025-02-19T21:54:35.4889869Z -----
2025-02-19T21:54:35.9171972Z /home/azureuser/.myagent/externals/node/12.18.0/bin/node: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.28' not found (required by /home/azureuser/myage
2025-02-19T21:54:36.7870192Z ##[warning]The agent operating system doesn't support Node20. Using Node16 instead. Please upgrade the operating system of the agent to remain compatible
2025-02-19T21:54:46.5283772Z Got service connection details for Azure App Service: 'my-sample-app-123'
2025-02-19T21:54:46.5283772Z
2025-02-19T21:54:46.5704162Z Package deployment using 'Deploy initiated'.
2025-02-19T21:54:46.5704162Z Deployment initiated successfully at https://my-sample-app-123-azurud7bcd7ye2hu.scm.westeurope-01.azurewebsites.net/api/deployments/9141371c-dfb2-4a47-9bc8-f8a
2025-02-19T21:55:31.6777362Z Successfully deployed web package to App Service.
2025-02-19T21:55:40.9320582Z Successfully updated App Service configuration details
2025-02-19T21:55:40.9320582Z Successfully updated deployment history at https://my-sample-app-123-azurud7bcd7ye2hu.scm.westeurope-01.azurewebsites.net/api/deployments/717400021492
2025-02-19T21:55:53.1071752Z App Service Application URL: https://my-sample-app-123-azurud7bcd7ye2hu.scm.westeurope-01.azurewebsites.net
2025-02-19T21:55:53.1071752Z -----
2025-02-19T21:55:53.1071752Z ##[section]Finishing: Azure App Service Deploy: my-sample-app-123

```

Practical Task 7: Use Azure Artifacts to Manage Dependencies

Requirements:

1. Create an Azure Artifacts Feed:

- In Azure DevOps, navigate to Artifacts and create a new feed named MyDependenciesFeed.

2. Publish Dependencies:

- Modify the build pipeline to include a task that publishes the application's dependencies (e.g., npm packages or NuGet packages) to the Azure Artifacts feed.

- Use commands like `npm publish` or `dotnet nuget push` in the pipeline.

3. Restore Dependencies:

- Update the build pipeline to restore dependencies from the Azure Artifacts feed during the build process.

4. Verify Dependency Management:

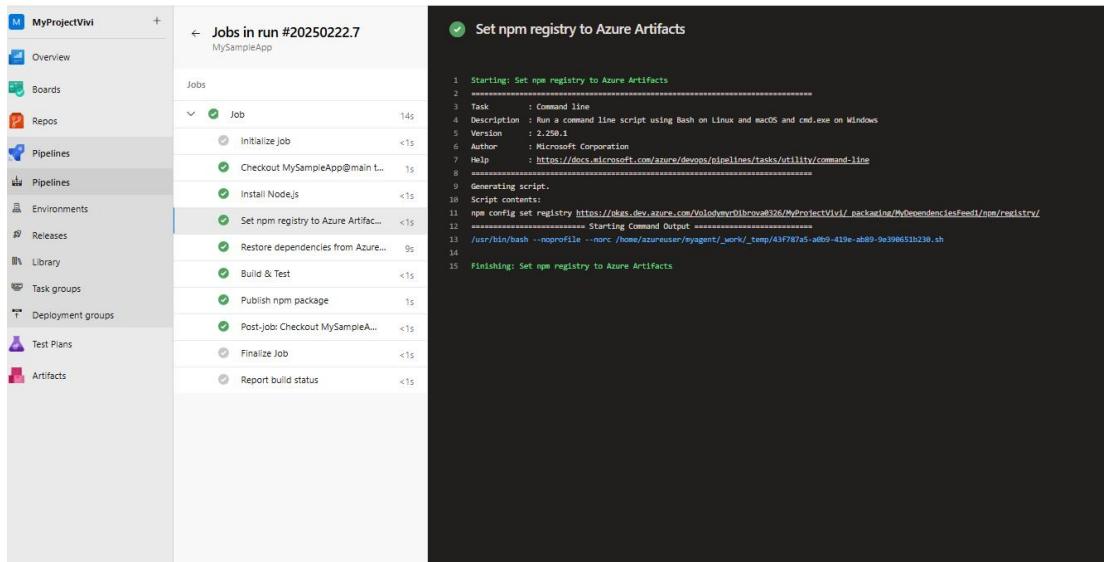
- Run the pipeline and verify that the dependencies are correctly published to and restored from the feed.

The image consists of two screenshots of the Azure DevOps interface.

The top screenshot shows the "Artifacts" page for the project "MyProjectVivi". A new feed named "MyDependenciesFeed1" has been created and is listed under "Active Feeds".

The bottom screenshot shows the "Pipelines" page for the project "MyProjectVivi". A pipeline named "MySampleApp" is selected. The "azure-pipelines.yml" file is open, displaying the YAML configuration for the pipeline. The configuration includes setting the npm registry to the newly created feed and publishing npm packages.

```
trigger:
- main
pool:
  name: MyLinux
  demands:
    - agent.name -equals myAgent1
variables:
  FEED_URL: "https://pkgs.dev.azure.com/VolodymyrDibrova0326/_packaging/MyDependenciesFeed1/npm/registry/"
steps:
- task: NodeTool@0
  inputs:
    versionSpec: '20.x'
    displayName: 'Install Node.js'
- script: npm config set registry ${FEED_URL}
  displayName: 'Set npm registry to Azure Artifacts'
- script: npm install
  displayName: 'Restore dependencies from Azure Artifacts'
- script: |
    npm run build
    npm test
  displayName: 'Build & Test'
- script: npm publish --registry=${FEED_URL}
  displayName: 'Publish npm package'
```



Practical Task 8: Deploy Infrastructure as Code (IaC) with Bicep Requirements:

1. Create a Bicep File:

- a. Create a Bicep file named **main.bicep** that defines the infrastructure for your application, including an Azure Web App and a SQL Database.

2. Create a New Pipeline:

- a. Set up a new pipeline in Azure DevOps that includes a stage for deploying the Bicep file.

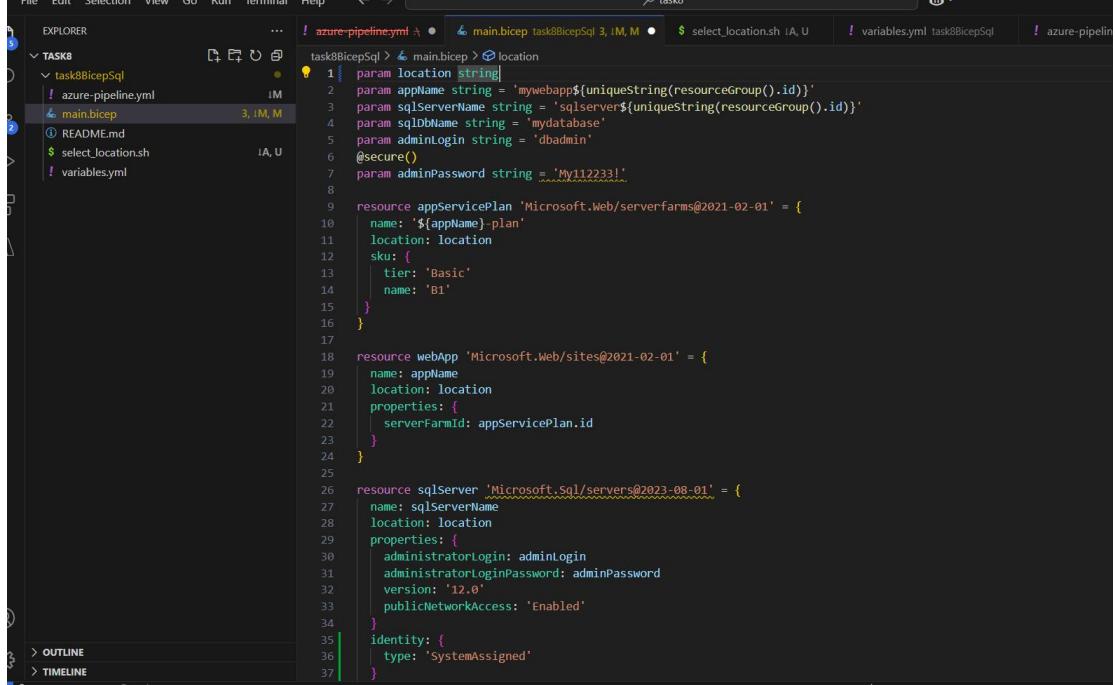
3. Deploy Using Azure CLI:

- a. Use the Azure CLI task in the pipeline to deploy the Bicep file.
- b. The command should look like:**az deployment group create --resource-group <YourResourceGroup> --template-file main.bicep**

4. Run the Pipeline:

- a. Save and run the pipeline to ensure that the infrastructure is provisioned correctly in Azure.
- b. Verify that the resources are created in the Azure portal.

1. Create a Bicep File



The screenshot shows the Azure DevOps interface with the 'task8BicepSql' pipeline selected. In the 'EXPLORER' view, there are several files: 'azur...yml', 'main.bicep', 'README.md', 'select_location.sh', and 'variables.yml'. The 'main.bicep' file is open in the code editor, displaying Bicep code for creating an App Service Plan, a Web App, and a SQL Server.

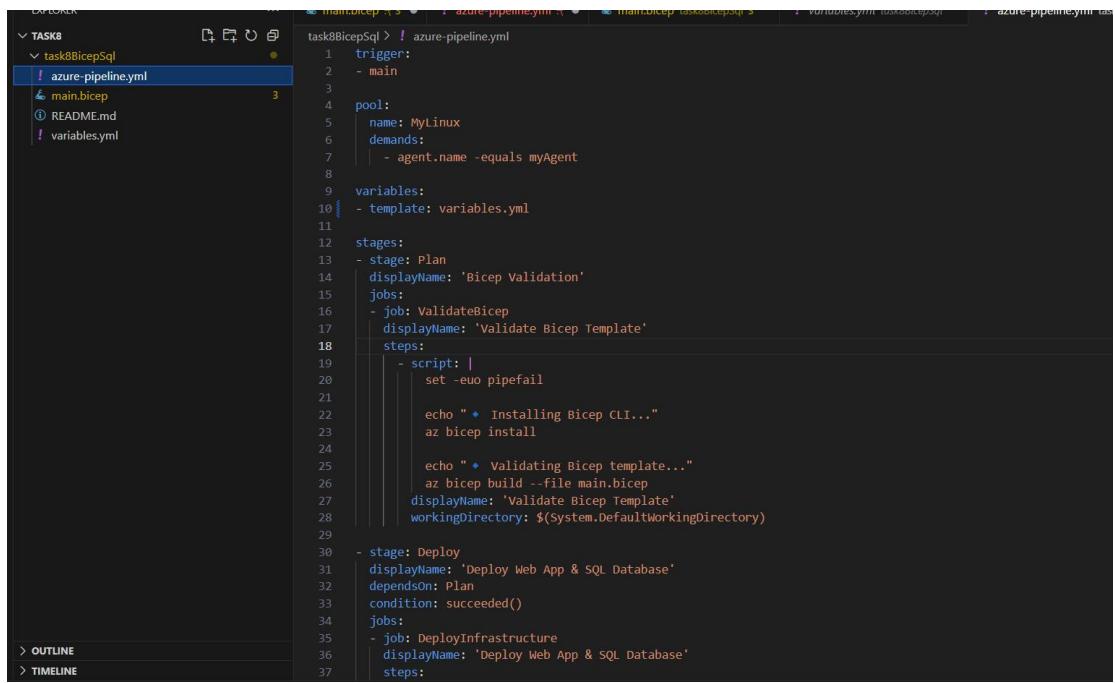
```
param location string
param appName string = 'mywebapp${uniqueString(resourceGroup().id)}'
param sqlServerName string = 'sqlserver${uniqueString(resourceGroup().id)}'
param sqlDbName string = 'mydatabase'
param adminLogin string = 'dbadmin'
@secure()
param adminPassword string = 'My112233!'

resource appServicePlan 'Microsoft.Web/serverfarms@2021-02-01' = {
  name: '${appName}-plan'
  location: location
  sku: {
    tier: 'Basic'
    name: 'B1'
  }
}

resource webApp 'Microsoft.Web/sites@2021-02-01' = {
  name: appName
  location: location
  properties: {
    serverFarmId: appServicePlan.id
  }
}

resource sqlServer 'Microsoft.Sql/servers@2023-08-01' = {
  name: sqlServerName
  location: location
  properties: {
    administratorLogin: adminLogin
    administratorLoginPassword: adminPassword
    version: '12.0'
    publicNetworkAccess: 'Enabled'
  }
  identity: {
    type: 'systemAssigned'
  }
}
```

Created pipeline



The screenshot shows the Azure DevOps interface with the 'task8BicepSql' pipeline selected. In the 'EXPLORER' view, there are files: 'azur...yml', 'main.bicep', 'README.md', and 'variables.yml'. The 'azur...yml' file is open in the code editor, showing the pipeline definition with a trigger, pool, variables, stages, and jobs.

```
trigger:
- main

pool:
  name: MyLinux
  demands:
  - agent.name -equals myAgent

variables:
- template: variables.yml

stages:
- stage: Plan
  displayName: 'Bicep Validation'
  jobs:
  - job: ValidateBicep
    displayName: 'Validate Bicep Template'
    steps:
    - script: |
        set -euo pipefail
        echo "• Installing Bicep CLI..."
        az bicep install

        echo "• Validating Bicep template..."
        az bicep build --file main.bicep
        displayName: 'Validate Bicep Template'
        workingDirectory: $(System.DefaultWorkingDirectory)

- stage: Deploy
  displayName: 'Deploy Web App & SQL Database'
  dependsOn: Plan
  condition: succeeded()
  jobs:
  - job: DeployInfrastructure
    displayName: 'Deploy Web App & SQL Database'
    steps:
```

```

variables:
  resourceGroupName: 'Volodymyr-Dibrova'
  location: 'East US'

```

Pushed to Repo

```

remote: Found 3 objects to send. (15 ms)
Unpacking objects: 100% (3/3), 760 bytes | 84.00 KiB/s, done.
PS E:\projects\terraformProj\DevopsService\taskService\task8> cd \task8BicepSql
PS E:\projects\terraformProj\DevopsService\taskService\task8> git add .
PS E:\projects\terraformProj\DevopsService\taskService\task8> git commit -m "bicep pipeline $ sql"
[main 07c41d5] bicep pipeline $ sql
 3 files changed, 69 insertions(+)
create mode 100644 azure-pipeline.yml
create mode 100644 main.bicep
create mode 100644 variables.yml
PS E:\projects\terraformProj\DevopsService\taskService\task8> git push origin master
error: src refspec master does not match any
error: failed to push some refs to 'https://dev.azure.com/VolodymyrDibrova0326/MyProjectVivi/_git/task8BicepSql'
PS E:\projects\terraformProj\DevopsService\taskService\task8> git branch
* main
PS E:\projects\terraformProj\DevopsService\taskService\task8> git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 1.22 KiB | 624.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Analyzing objects... (4/4) (5 ms)
remote: Validating commits... (1/1) done (0 ms)
remote: Storing packfile... done (27 ms)
remote: Storing index... done (32 ms)

```

Created a program to search for available regions

```

#!/bin/bash
locations=("eastus" "centralus" "westeurope" "northeurope" "westus")
echo "Checking available regions..."
selectedLocation=""
for loc in "${locations[@]}"
do
  echo " Checking region: $loc"
  available=$(az sql db list-editions -l "$loc" --query "[?serviceObjective=='Basic'].serviceObjective" -o tsv)
  echo " Azure CLI response: $available"
  if [[ -n "$available" ]]; then
    selectedLocation=$loc
    echo " Selected region: $selectedLocation"
    break
  fi
done
if [[ -z "$selectedLocation" ]]; then
  echo " No regions available! Try another set or check quotas."
  exit 1
fi
echo "##vso[task.setvariable variable=selectedLocation]$selectedLocation"

```

```

2025-02-20T20:15:01.6339276Z
2025-02-20T20:15:01.6339634Z Python (Linux) 3.11.8 (main, Apr 24 2024, 04:14:33) [GCC 7.5.0]
2025-02-20T20:15:01.6339790Z
2025-02-20T20:15:01.6340100Z Legal docs and information: aka.ms/AzureClilegal
2025-02-20T20:15:01.6340275Z
2025-02-20T20:15:01.6340362Z
2025-02-20T20:15:01.6340689Z Setting AZURE_CONFIG_DIR env variable to: /home/azureuser/myagent/_work/_temp/.azclitask
2025-02-20T20:15:01.6341091Z Setting active cloud to: AzureCloud
2025-02-20T20:15:01.6341414Z [command]/usr/bin/az cloud set -n AzureCloud
2025-02-20T20:15:04.823532Z [command]/usr/bin/az login --service-principal -u *** --tenant 8d1157bb-1f96-415f-824b-ab0a29485d7d --allow-no-subscriptions --federated-token ***
2025-02-20T20:15:06.2435532Z
2025-02-20T20:15:06.2436239Z
2025-02-20T20:15:06.2436595Z
2025-02-20T20:15:06.2437010Z
2025-02-20T20:15:06.2437439Z
2025-02-20T20:15:06.2437811Z
2025-02-20T20:15:06.2438155Z
2025-02-20T20:15:06.2438511Z
2025-02-20T20:15:06.2438861Z
2025-02-20T20:15:06.2439234Z
2025-02-20T20:15:06.2439571Z [command]/usr/bin/az account set --subscription 9a6ae428-d8c3-44fe-bdf2-4e08593901a0
2025-02-20T20:15:06.2440070Z
2025-02-20T20:15:06.2440422Z
2025-02-20T20:15:06.2440743Z
2025-02-20T20:15:06.2441038Z
2025-02-20T20:15:06.2441138Z ]
2025-02-20T20:15:06.7915138Z [command]/usr/bin/bash /home/azureuser/myagent/_work/_temp/azureclitaskscript1740082501016.sh
2025-02-20T20:15:06.8884019Z Checking available regions...
2025-02-20T20:15:06.8822086Z
2025-02-20T20:15:08.2479200Z Azure CLI response:
2025-02-20T20:15:08.2480329Z Checking region: eastus
2025-02-20T20:15:09.5460214Z Azure CLI response:
2025-02-20T20:15:09.5461231Z Checking region: centralus
2025-02-20T20:15:11.3728467Z Azure CLI response:
2025-02-20T20:15:11.3729337Z Checking region: northeurope
2025-02-20T20:15:13.9699956Z Azure CLI response:
2025-02-20T20:15:13.9701874Z Checking region: westus
2025-02-20T20:15:15.4235815Z Azure CLI response:
2025-02-20T20:15:15.4236959Z ✘ No regions available! Try another set or check quotas.
2025-02-20T20:15:15.4269608Z ##[error]Script failed with exit code: 1
2025-02-20T20:15:15.4289604Z [command]/usr/bin/az account clear
2025-02-20T20:15:16.3368236Z ##[section]Finishing: Checking available regions

```

My subscription has a quota limit in the selected region for the App Service Plan

```

PS E:\projects\terraformProj\DevopsService\taskService\task8\task8BicepSql> az deployment group create --resource-group Volodymyr-Dibrova --template-file main.bicep
E:\projects\terraformProj\DevopsService\taskService\task8\task8BicepSql\main.bicep(7,28) : Warning secure-parameter-default: Secure parameters should not have hardcoded default (except for empty or nullified()). [https://aka.ms/bicep/interpolated-parameters#default]
E:\projects\terraformProj\DevopsService\taskService\task8\task8BicepSql\main.bicep(26,20) : Warning BC0081: Resource type "Microsoft.Sql/servers@2023-08-01" does not have types available. Bicep is unable to validate resource properties prior to deployment, but this will not block the resource from being deployed. [https://aka.ms/bicep/core-diagnostics#BCP0081]
E:\projects\terraformProj\DevopsService\taskService\task8\task8BicepSql\main.bicep(41,22) : Warning BC0081: Resource type "Microsoft.Sql/servers/databases@2023-08-01" does not have types available. Bicep is unable to validate resource properties prior to deployment, but this will not block the resource from being deployed. [https://aka.ms/bicep/core-diagnostics#BCP0081]

{
    "id": "/subscriptions/9a6ae428-d8c3-44fe-bdf2-4e08593901a0/resourceGroups/Volodymyr-Dibrova/providers/Microsoft.Resources/deployments/main",
    "location": null,
    "name": "main",
    "properties": {
        "correlationId": "d7e53c5f-9bc1-49ae-a237-2c8843343162",
        "debugSetting": "null",
        "dependencies": [
            {
                "dependsOn": [
                    {
                        "id": "/subscriptions/9a6ae428-d8c3-44fe-bdf2-4e08593901a0/resourceGroups/Volodymyr-Dibrova/providers/Microsoft.Web/serverfarms/mywebapps3yd6pzpxyqjw-plan",
                        "resourceGroup": "Volodymyr-Dibrova",
                        "resourceName": "mywebapps3yd6pzpxyqjw-plan",
                        "resourceType": "Microsoft.Web/serverfarms"
                    }
                ],
                "id": "/subscriptions/9a6ae428-d8c3-44fe-bdf2-4e08593901a0/resourceGroups/Volodymyr-Dibrova/providers/Microsoft.Web/sites/mywebapps3yd6pzpxyqjw",
                "resourceGroup": "Volodymyr-Dibrova",
                "resourceName": "mywebapps3yd6pzpxyqjw",
                "resourceType": "Microsoft.Web/sites"
            },
            {
                "dependsOn": [
                    {
                        "id": "/subscriptions/9a6ae428-d8c3-44fe-bdf2-4e08593901a0/resourceGroups/Volodymyr-Dibrova/providers/Microsoft.Sql/servers/3yd6pzpxyqjw",
                        "resourceGroup": "Volodymyr-Dibrova",
                        "resourceName": "sqlservers3yd6pzpxyqjw",
                        "resourceType": "Microsoft.Sql/servers"
                    }
                ],
                "id": "/subscriptions/9a6ae428-d8c3-44fe-bdf2-4e08593901a0/resourceGroups/Volodymyr-Dibrova/providers/Microsoft.Sql/servers/sqlservers3yd6pzpxyqjw/databases/mydatabase",
                "resourceGroup": "Volodymyr-Dibrova",
                "resourceName": "sqlservers3yd6pzpxyqjw/mydatabase",
                "resourceType": "Microsoft.Sql/databases"
            }
        ]
    }
}

```

The screenshot shows the Microsoft Azure Resource Group Overview page for the resource group 'Volodymyr-Dibrova'. The page displays a list of resources with columns for Name, Type, and Location. The resources listed are:

Name	Type	Location
mydatabase (sqlservers3yld6pzhoyqjw/mydatabase)	SQL database	West Europe
myVM	Virtual machine	East US
myVM-ip	Public IP address	East US
myVM-nic	Network interface	East US
myVM-nsg	Network security group	East US
myVM-vnet	Virtual network	East US
myVM_disk_1_c5f4b271e944b5dacd2b69fb65ceca3	Disk	East US
mywebappssyld6pzhoyqjw	App Service	West Europe
mywebappssyld6pzhoyqjw-plan	App Service plan	West Europe

Practical Task 9: Implement Infrastructure as Code (IaC) with Terraform Requirements:

1. Create a Terraform Configuration:

- Create a Terraform configuration file (`main.tf`) that defines the same infrastructure as in Task 8 (e.g., Azure Web App and SQL Database).

2. Set Up a New Azure Pipeline:

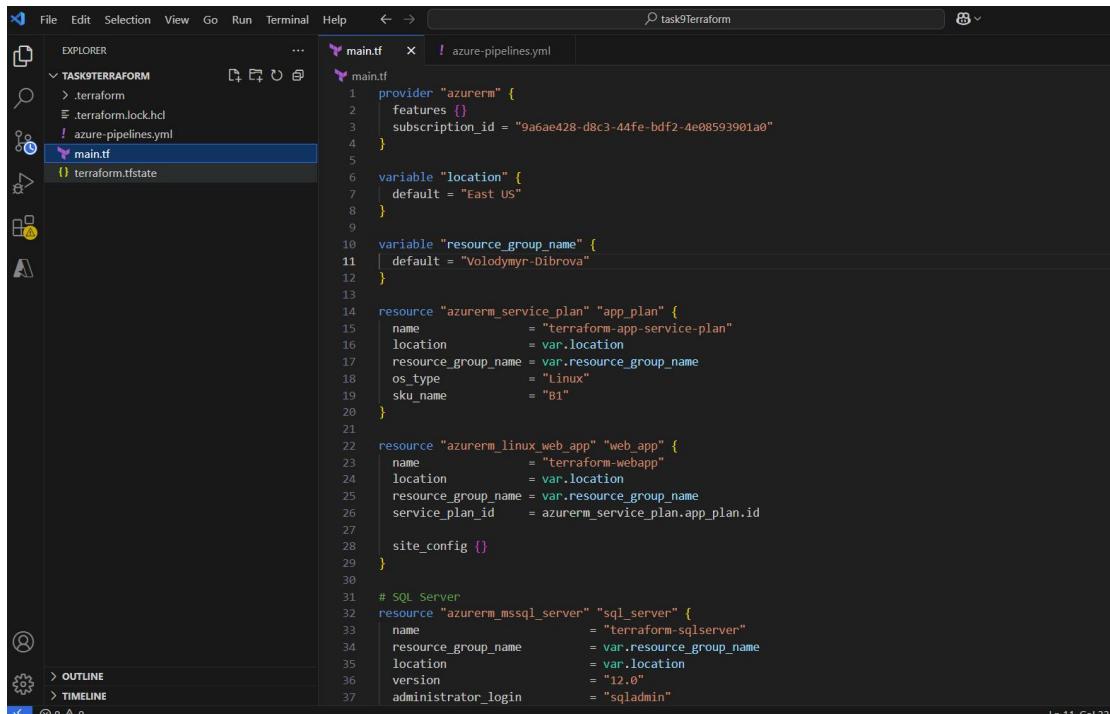
- Create a new Azure Pipeline that runs Terraform commands.
- Use the following steps in the pipeline:
`terraform init` to initialize the Terraform working directory.
`terraform plan` to create an execution plan.
`terraform apply -auto-approve` to apply the changes.

3. Configure Triggers:

- Set up the pipeline to trigger on changes to the Terraform files in the repository.

4. Run the Pipeline:

- Save and run the pipeline to provision the infrastructure in Azure.
- Verify that the resources are created as expected.



The screenshot shows the Visual Studio Code interface with the Terraform extension installed. The Explorer sidebar on the left lists files: .terraform, .terraform.lock.hcl, azure-pipelines.yml, main.tf, and terraform.tfstate. The main editor area displays the contents of main.tf:

```
provider "azurerm" {
  features {}
  subscription_id = "9a6ae428-d8c3-44fe-bdf2-4e08593901a0"
}

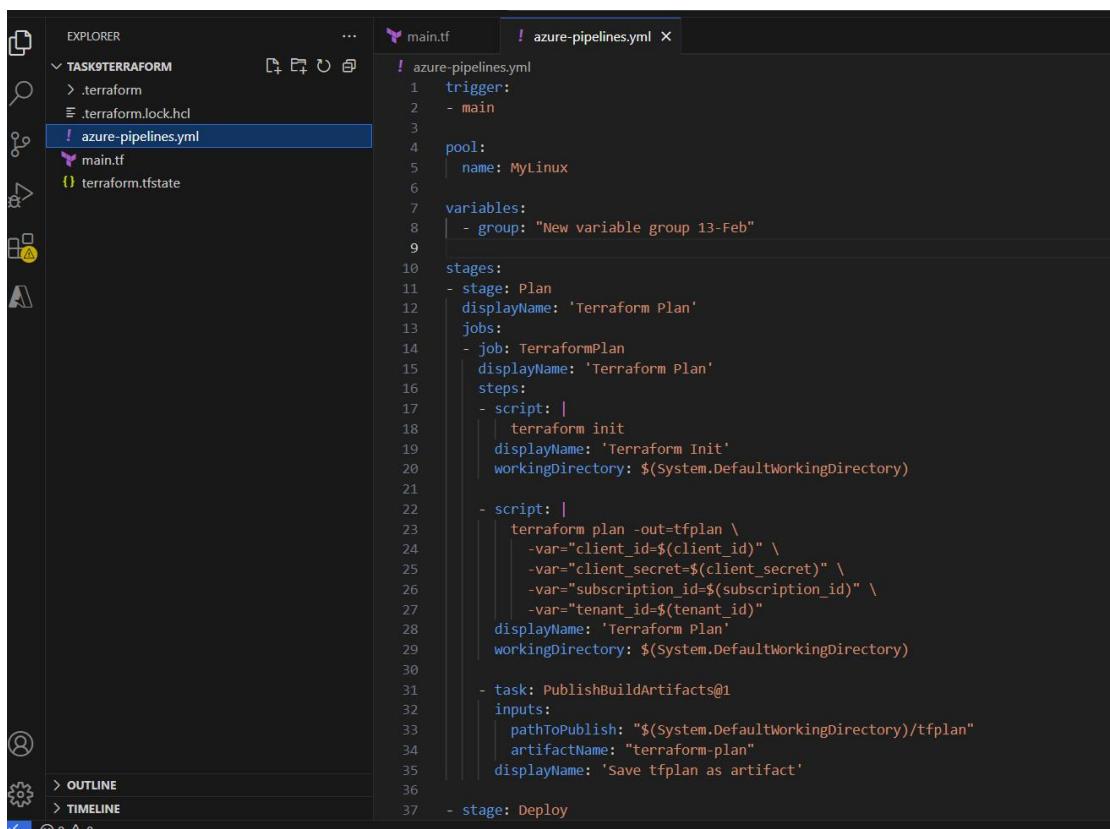
variable "location" {
  default = "East US"
}

variable "resource_group_name" {
  default = "Volodymyr-Dibrova"
}

resource "azurerm_service_plan" "app_plan" {
  name          = "terraform-app-service-plan"
  location      = var.location
  resource_group_name = var.resource_group_name
  os_type       = "Linux"
  sku_name      = "B1"
}

resource "azurerm_linux_web_app" "web_app" {
  name          = "terraform-webapp"
  location      = var.location
  resource_group_name = var.resource_group_name
  service_plan_id = azurerm_service_plan.app_plan.id
  site_config {}
}

# SQL Server
resource "azurerm_mssql_server" "sql_server" {
  name          = "terraform-sqlserver"
  resource_group_name = var.resource_group_name
  location      = var.location
  version        = "12.0"
  administrator_login = "sqladmin"
}
```



The screenshot shows the Visual Studio Code interface with the Azure Pipelines extension installed. The Explorer sidebar on the left lists files: .terraform, .terraform.lock.hcl, azure-pipelines.yml, main.tf, and terraform.tfstate. The main editor area displays the contents of azure-pipelines.yml:

```
trigger:
- main

pool:
- name: MyLinux

variables:
- group: "New variable group 13-Feb"

stages:
- stage: Plan
  displayName: 'Terraform Plan'
  jobs:
    - job: TerraformPlan
      displayName: 'Terraform Plan'
      steps:
        - script: |
            terraform init
            displayName: 'Terraform Init'
            workingDirectory: $(System.DefaultWorkingDirectory)

        - script: |
            terraform plan -out=tfplan \
            -var="client_id=$(client_id)" \
            -var="client_secret=$(client_secret)" \
            -var="subscription_id=$(subscription_id)" \
            -var="tenant_id=$(tenant_id)"
            displayName: 'terraform Plan'
            workingDirectory: $(System.DefaultWorkingDirectory)

    - task: PublishBuildArtifacts@1
      inputs:
        pathToPublish: "$(System.DefaultWorkingDirectory)/tfplan"
        artifactName: "terraform-plan"
        displayName: 'Save tfplan as artifact'

- stage: Deploy
```

```
you run "terraform apply" now.
PS E:\projects\terraformProj\DevopsService\taskService\task9Terraform> terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
+ create

Terraform will perform the following actions:

# azurerm_linux_web_app.web_app will be created
+ resource "azurerm_linux_web_app" "web_app" {
    + client_affinity_enabled          = false
    + client_certificate_enabled        = false
    + client_certificate_mode          = "Required"
    + custom_domain_verification_id    = (sensitive value)
    + default_hostname                 = (known after apply)
    + enabled                          = true
    + ftp_publish_basic_authentication_enabled = true
    + hosting_environment_id           = (known after apply)
    + https_only                       = false
    + id                               = (known after apply)
    + key_vault_reference_identity_id = (known after apply)
    + kind                             = (known after apply)
    + location                         = "eastus"
    + name                            = "terraform-webapp"
    + outbound_ip_address_list         = (known after apply)
    + outbound_ip_addresses            = (known after apply)
    + possible_outbound_ip_address_list = (known after apply)
    + possible_outbound_ip_addresses   = (known after apply)
    + public_network_access_enabled    = true
    + resource_group_name              = "Volodymyr-Dibrova"
    + service_plan_id                 = (known after apply)
    + site_credential                  = (sensitive value)
    + webdeploy_publish_basic_authentication_enabled = true
    + zip_deploy_file                 = (known after apply)

    + site_config {
        + always_on                      = true
        + container_registry_use_managed_identity = false
        + default_documents               = (known after apply)
    }
}
```

```
azurerm_mssql_database.sql_db: Still creating... [1m58s elapsed]
azurerm_mssql_database.sql_db: Still creating... [2m0s elapsed]
azurerm_mssql_database.sql_db: Still creating... [2m18s elapsed]
azurerm_mssql_database.sql_db: Still creating... [2m20s elapsed]
azurerm_mssql_database.sql_db: Creation complete after 2m27s [id=subscriptions/9a6ae428-d8c3-44fe-bdf2-4e08593901a0/resourceGroups/Volodymyr-Dibrova/providers/Microsoft.Sql/servers/terraform-sqlserver/databases/terraform-db]

Error: creating App Service Plan (Subscription: "9a6ae428-d8c3-44fe-bdf2-4e08593901a0"
Resource Group Name: "Volodymyr-Dibrova"
Server Farm Name: "terraform-app-service-plan"): performing CreateOrUpdate: unexpected status 401 (401 Unauthorized) with response: {"Code":"Unauthorized","Message":"This region has quota of 0 instances for your subscription. Try selecting different region or SKU."}, {"Target":null,"Details":[{"Message":"This region has quota of 0 instances for your subscription. Try selecting different region or SKU."}], "Code":"Unauthorized"}, {"ErrorEntity":{"ExtendedCode":"52039","MessageTemplate":"{}"}, "Parameters":[{"Message":"This region has quota of 0 instances for your subscription."}], "Code":"Unauthorized"}, {"Message":"This region has quota of 0 instances for your subscription. Try selecting different region or SKU."}], "Innererror":null}

  with azurerm_service_plan.app_plan,
  on main.tf line 14, in resource "azurerm_service_plan" "app_plan":
  14: resource "azurerm_service_plan" "app_plan" {
```

creating App Service Plan (Subscription: "9a6ae428-d8c3-44fe-bdf2-4e08593901a0"
Resource Group Name: "Volodymyr-Dibrova"
Server Farm Name: "terraform-app-service-plan"): performing CreateOrUpdate: unexpected status 401 (401 Unauthorized) with response: {"Code":"Unauthorized","Message":"This region has quota of 0 instances for your subscription. Try selecting different region or SKU."}, {"Target":null,"Details":[{"Message":"This region has quota of 0 instances for your subscription. Try selecting different region or SKU."}], "Code":"Unauthorized"}, {"ErrorEntity":{"ExtendedCode":"52039","MessageTemplate":"{}"}, "Parameters":[{"Message":"This region has quota of 0 instances for your subscription."}], "Code":"Unauthorized"}, {"Message":"This region has quota of 0 instances for your subscription. Try selecting different region or SKU."}], "Innererror":null}

Microsoft Azure

Home > Volodymyr-Dibrova Resource group

Search resources, services, and docs (G+)

Copilot

[Create](#) [Manage view](#) [Delete resource group](#) [Refresh](#) [Export to CSV](#) [Open query](#) | [Assign tags](#) [Move](#) [Delete](#) [Export template](#) [Open in mobile](#)

Overview

Essentials

Subscription (move) : Azure subscription 1 Deployments : 5 Succeeded
Subscription ID : 9a6ae428-d8c3-44fe-bdf2-4e08593901a0 Location : East US
Tags (edit) : Add tags

Resources Recommendations (22)

Filter for any field... Type equals all Location equals all Add filter

Showing 1 to 12 of 12 records. Show hidden types

Name	Type	Location
mydatabase (sqlservers3y6pzhyqjw/mydatabase)	SQL database	West Europe
myVM	Virtual machine	East US
myVM-ip	Public IP address	East US
myVM-nic	Network Interface	East US
myVM-nsg	Network security group	East US
myVM-vnet	Virtual network	East US
myVM_disk1_c55fb4b271e944b5dacd3b69fb65ceca3	Disk	East US
mywebapps3y6pzhyqjw	App Service	West Europe
mywebapps3y6pzhyqjw-plan	App Service plan	West Europe

< Previous Page 1 of 1 Next >

```

1 Starting: Select best Azure region
2 -----
3 Task : Command line
4 Description : Run a command line script using Bash on Linux and macOS and cmd.exe on Windows
5 Version : 2.250.1
6 Author : Microsoft Corporation
7 Help : https://docs.microsoft.com/azure/devops/pipelines/tasks/utility/command-line
8 -----
9 /home/azureuser/myagent/externals/node20_1/bin/node
10 /home/azureuser/myagent/externals/node20_1/bin/node: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.28' not found (required by /home/azureuser/myagent/externals/node20_1/bin/node)
11 ##[warning]The agent operating system doesn't support node20. Using Node10 instead. Please upgrade the operating system of the agent to remain compatible.
12 Generating script.
13 ===== Starting Command Output =====
14 /bin/bash -noprompt --nrc /home/azureuser/myagent/_work/_temp/e8134e43-5541-4ae7-9d1e-4ecc54b2cf6d.sh
15 ● Checking available regions...
16 ● Checking region: eastus
17 ● Azure CLI response:
18 ● Checking region: centralus
19 ● Azure CLI response:
20 ● Checking region: westeurope
21 ● Azure CLI response:
22 ● Checking region: northeurope
23 ● Azure CLI response:
24 ● Checking region: westus
25 ● Azure CLI response:
26 ✘ No regions available! Try another set or check quotas.
27 ##[error]Batch exited with code '-1'.
28 Finishing: Select best Azure region

```

Practical Task 10: Integrate Azure Test Plans with CI/CD Pipelines

Requirements:

1. Create a Test Plan:

- Navigate to Azure Test Plans and create a new test plan that includes both manual and automated tests for your application.

2. Modify the Build Pipeline:

- Update the build pipeline to include a testing stage that runs automated tests using a testing framework (e.g., NUnit for .NET or Jest for Node.js).
- Add a task in the pipeline to publish the test results.

3. Configure Release Pipeline:

- Modify the release pipeline to deploy the application only if all tests pass.
- Use conditions in the release pipeline to check for successful test results.

4. Run the Pipeline:

- Trigger the pipeline and verify that it runs the tests and reports the results back to Azure Test Plans.
- Ensure that any failed tests block the deployment process.

Azure DevOps VolodymyrDibrova0326 / MyProjectVivi / Repos / Files / testJest

MyProjectVivi	testJest	main
Overview	azure-pipelines.yml	/ sum.js
Boards	package.json	sum.js
Repos	README.md	Contents History Compare Blame
Files	sum.js	1 function sum(a, b) { 2 return a + b; 3 } 4 5 module.exports = sum; 6
Commits		
Pushes		
Branches		
Tags		
Pull requests		

```

sum.test.js

Contents History Compare Blame

1 const sum = require("./sum");
2
3 test("adds 1 + 2 to equal 3", () => {
4   expect(sum(1, 2)).toBe(3);
5 });
6

```

```

azure-pipelines.yml

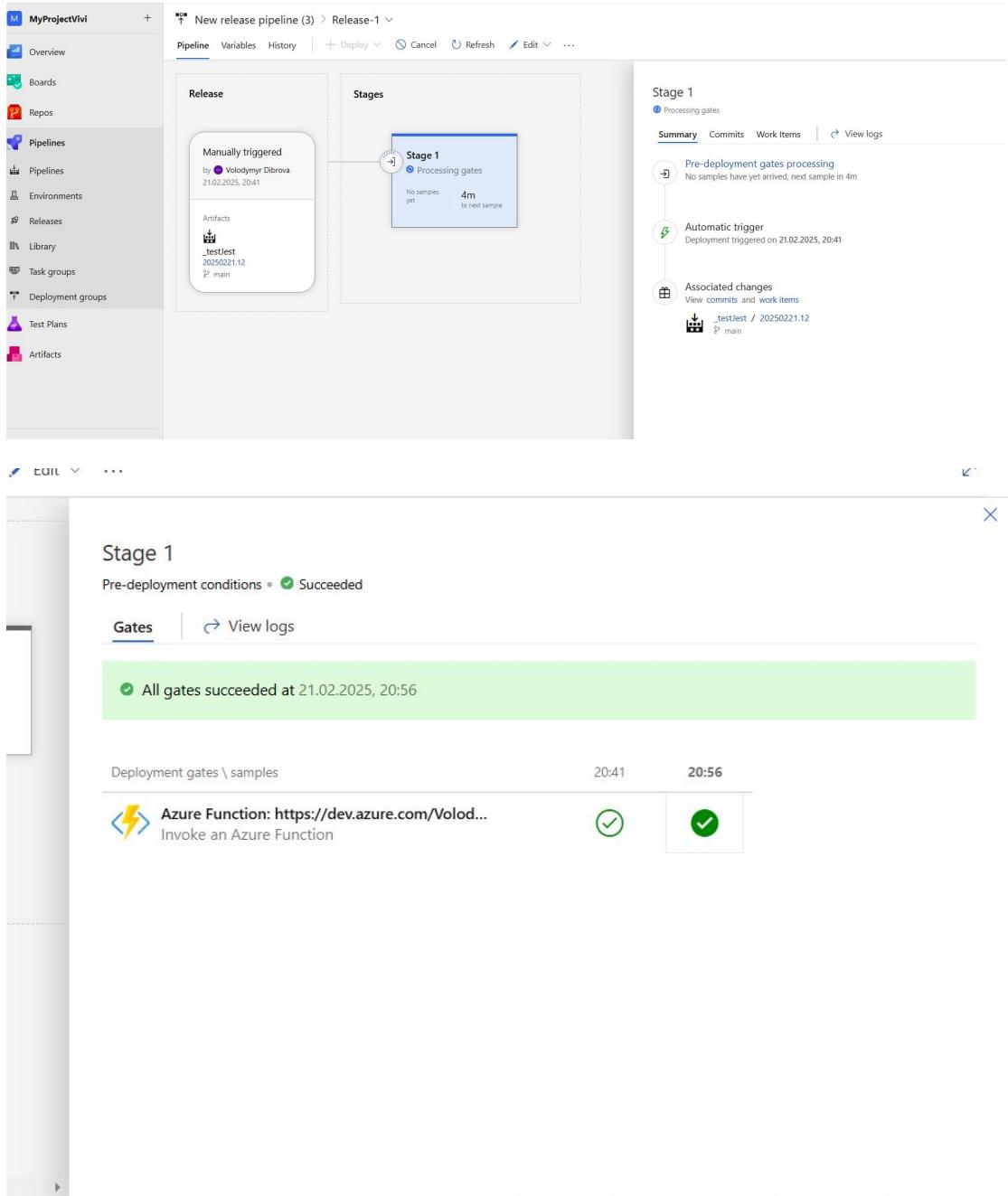
Contents History Compare Blame

1 trigger:
2 - main
3
4 pool:
5   name: MyLinux
6   demands:
7     - agent.name -equals myAgent1
8
9 steps:
10 - task: NodeTool@0
11   inputs:
12     versionSpec: '20.x'
13     displayName: 'Install Node.js 20'
14
15 - script:
16   npm install
17   npm test
18   displayName: 'Run Jest Tests'
19
20 - task: PublishTestResults@2
21   inputs:
22     testResultsFiles: '**/jest-report.xml'
23     testRunTitle: 'Jest Tests'
24     condition: always()
25

```

Run 3 - Jest Tests

Outcome	Test Case Title	Priority	Duration	Owner	Configuration	Machine name	Error message
Passed	adds 1 + 2 to equal 3		0	0:00:00.003		None	



Practical Task 11: Full CI/CD Pipeline with Infrastructure as Code, Testing, and Artifact Management

Requirements:

1. Design the Application Architecture:

- a. Create a simple application that consists of multiple components (e.g., a frontend application, a backend API, and a database).

2. Create a Bicep File for Infrastructure:

- a. Create a Bicep file (main.bicep) that provisions all necessary resources for the application, including:

- i. An Azure App Service for the frontend.

- ii. An Azure Function or Azure App Service for the backend API.
 - iii. An Azure SQL Database or Cosmos DB for data storage.
- b. Ensure that the Bicep file includes networking components, such as a Virtual Network and Subnets if needed.
- 3. Create Terraform Configuration:**
- a. Create a Terraform configuration (`main.tf`) that defines the same infrastructure as the Bicep file.
 - b. Ensure that the Terraform configuration can be used to provision the infrastructure independently.
- 4. Set Up Azure Artifacts:**
- a. Create an Azure Artifacts feed to store the application's packages (e.g., npm packages for the frontend and NuGet packages for the backend).
 - b. Modify the build pipelines to publish the application's packages to the Azure Artifacts feed.
- 5. Create CI/CD Pipelines:**
- a. Create two separate pipelines: one for the frontend and one for the backend.
- 6. Frontend Pipeline:**
- a. The pipeline should:
 - i. Trigger on changes to the frontend code in the repository.
 - ii. Build the frontend application and run unit tests.
 - iii. Publish the build artifacts to Azure Artifacts.
 - iv. Deploy the frontend application to the Azure App Service using the Bicep file.
- 7. Backend Pipeline:**
- a. The pipeline should:
 - i. Trigger on changes to the backend code in the repository.
 - ii. Build the backend API and run unit tests.
 - iii. Publish the backend API package to Azure Artifacts.
 - iv. Deploy the backend API to Azure using the Bicep file.
- 8. Integrate Azure Test Plans:**
- a. Create a comprehensive test plan in Azure Test Plans that includes both manual and automated tests for both the frontend and backend.
 - b. Update both pipelines to include a testing stage that runs automated tests and publishes the results to Azure Test Plans.
- 9. Implement Continuous Monitoring:**
- a. Set up Application Insights for both the frontend and backend applications to monitor performance and errors.
 - b. Configure alerts for critical issues that may arise in the production environment.
- 10. Run the Full CI/CD Process:**
- a. Trigger both pipelines by making changes to the codebase.
 - b. Verify that the entire CI/CD process runs smoothly, from code commit to deployment.
 - c. Check the Azure portal to ensure all resources are provisioned correctly and that the applications are running as expected.

My project structure

fullstack-azure-app

- backend
- frontend
- infrastructure
- azure-pipelines-backend.yml
- .npmrc
- azure-pipelines-frontend.yml
- README.md
- variables.yml

Files

Name	Last change	Commits
backend	46m ago	f1e8dce4 Updated package.json Volodymyr Dibrova
frontend	2h ago	31b8e678 Updated package.json Volodymyr Dibrova
infrastructure	Yesterday	61c2d1fd Updated main.bicep Volodymyr Dibrova
azure-pipelines-backend.yml	11m ago	06b117b4 Updated azure-pipelines-backend.yml Volodymyr Dibrova
.npmrc	Yesterday	c44ea0d4 Added .npmrc Volodymyr Dibrova
azure-pipelines-frontend.yml	2h ago	8c7cfc78 Updated azure-pipelines-frontend.yml Volodymyr Dibrova
README.md	пятница	d299cc2e Added README.md Volodymyr Dibrova
variables.yml	Yesterday	062b262f Renamed variables.yml to variables.yml Volodymyr Dibrova

Introduction

TODO: Give a short introduction of your project. Let this section explain the objectives or the motivation behind this project.

1. Created a simple application that consists of multiple components a frontend application, a backend API, and a database

frontend

```

trigger:
- main

pool:
  name: MyLinux
  demands:
    - windows-2019
    - visual-studio-code-latest

steps:
- task: NodeTool@0
  inputs:
    versionSpec: '14.x'
  displayName: 'Install Node.js'

- script:
  - npm ci
  displayName: 'Install dependencies and build frontend'

- task: AzureWebApp@1
  inputs:
    appType: 'Function App for Azure'
    appName: 'Frontend/hello-azure'
    deploymentMethod: 'Copy app to Frontend'
    workingDirectory: '$(Build.SourcesDirectory)'

- script:
  - az webapp up
  displayName: 'Deploy Frontend/hello-azure'

- task: PublishBuildArtifacts@1
  inputs:
    PathtoPublish: '$(Build.ArtifactStagingDirectory)/Frontend/hello-azure'
    ArtifactName: 'Frontend/hello-azure'
  displayName: 'Archive Frontend build'

```

Jobs in run #20250222.21

fullstack-azure-app

Jobs

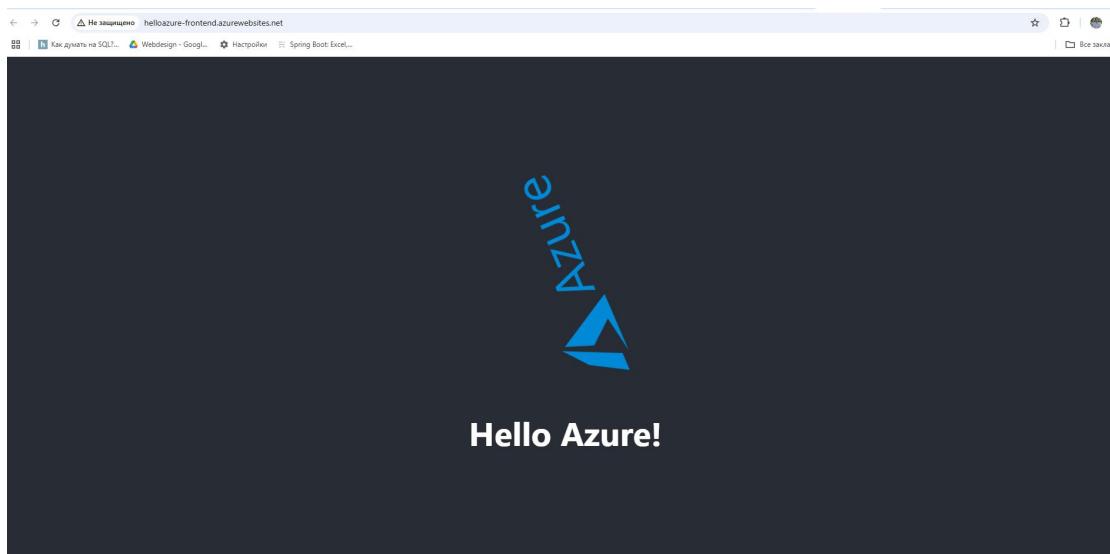
- Job
- Initialize job
- Checkout fullstack-azure-app@main
- Install Nodejs
- Check if package.json exists before build
- Install dependencies and build front end
- Check build output
- Copy .npmrc to frontend
- Test package creation
- Publish to Azure Artifacts
- Install zip utility
- Archive front end build
- Deploy Frontend to Azure
- Post-job: Checkout fullstack-azu...
- Finalize Job
- Report build status

Publish to Azure Artifacts

```

Starting: Publish to Azure Artifacts
-----
Task : Command line
Description : Run a command line script using Bash on Linux and macOS and cmd.exe on Windows
Version : 2.250.1
Author : Microsoft Corporation
Help : https://docs.microsoft.com/azure/devops/pipelines/tasks/utility/command-line
-----
Generating script.
Script contents:
npm publish --registry=https://pkgs.dev.azure.com/VolodymyrDibrova/_packaging/MyDependenciesFeed/npm/registry/
=====
Starting Command Output =====
/usr/bin/bash --non-interactive /home/vsts/work/_temp/122db5ff-e638-4578-b7ea-32cde6795ce.sh
npm notice
npm notice ⚡ mydependenciesfeed@Hello-Azure@0.1.3
npm notice └── Tarball Content
npm notice   7 npm notice   3.4kb README.md
npm notice   7 npm notice   29.8kb mydependenciesfeed@Hello-Azure-0.1.3.tgz
npm notice   7 npm notice   1.0kb package.json
npm notice   7 npm notice   3.9kb public/favicon.ico
npm notice   7 npm notice   1.7kb public/index.html
npm notice   7 npm notice   5.3kb public/logo192.png
npm notice   7 npm notice   9.7kb public/logo512.png
npm notice   7 npm notice   402B public/manifest.json
npm notice   7 npm notice   67B public/robots.txt
npm notice   7 npm notice   564B src/App.css
npm notice   7 npm notice   299B src/App.js
npm notice   7 npm notice   246B src/App.test.js
npm notice   7 npm notice   7.5kB src/azure.svg
npm notice   7 npm notice   366B src/index.css
npm notice   7 npm notice   5kB src/index.js
npm notice   7 npm notice   6kB src/logo.svg
npm notice   7 npm notice   342B reportWebVitals.js
npm notice   7 npm notice   241B src/setupTests.js
npm notice Tarball Details

```



backend



The screenshot shows the Azure DevOps interface with the following details:

- Project:** MyProjectV11
- Pipeline:** fullstack-azure-app
- YAML File:** azure-pipelines-backend.yml
- Contents:**

```
trigger:
  - main
  - master
  - tags

name: MyV11

demands:
  - agent.name equals myagent1
  - tags

steps:
  - task: NodeTool@0
    inputs:
      versionSpec: '20.x'
    displayName: 'Install Node.js'

  - script:
    inputs:
      workingDirectory: 'backend'
    steps:
      - task: Extract@0
        inputs:
          archiveFile: 'node-v14.15.4-linux-x64.tar.gz'
          destination: 'backend'
        displayName: 'Extract Node.js archive'

      - task: CopyFiles@2
        inputs:
          targetFolder: '$(System.DefaultWorkingDirectory)\backend\node_modules'
          contents:
            - '**/*'
        displayName: 'Copy node modules to backend'

      - task: CopyFiles@2
        inputs:
          targetFolder: '$(System.DefaultWorkingDirectory)\backend'
          contents:
            - '**/*'
        displayName: 'Copy .spec files to backend'

      - script:
        inputs:
          workingDirectory: 'backend'
        steps:
          - task: RunShellScript@0
            inputs:
              script: |
                cp -r ./spec backend/
                cp -r ./spec backend/
            displayName: 'Copy .spec to backend'

      - script:
        inputs:
          workingDirectory: 'backend'
        steps:
          - task: RunShellScript@0
            inputs:
              script: |
                rm -rf ./spec
                rm -rf ./spec
            displayName: 'Remove .spec files'

      - task: Npm@1
        inputs:
          command: install
          workingDirectory: 'backend'
        displayName: 'Install dependencies and build backend'

      - script:
        inputs:
          workingDirectory: 'backend'
        steps:
          - task: RunShellScript@0
            inputs:
              script: |
                cp -r ./spec backend/
                cp -r ./spec backend/
            displayName: 'Check backend directory after install'

      - script:
        inputs:
          workingDirectory: 'backend'
        steps:
          - task: RunShellScript@0
            inputs:
              script: |
                cp -r ./spec backend/
                cp -r ./spec backend/
            displayName: 'Copy .spec to backend'

      - script:
        inputs:
          workingDirectory: 'backend'
        steps:
          - task: RunShellScript@0
            inputs:
              script: |
                rm -rf ./spec
                rm -rf ./spec
            displayName: 'Remove .spec files'

      - task: Npm@1
        inputs:
          command: publish
          publishJUnitResults: true
          workingDirectory: '$(Build.SourcesDirectory)/backend'
        displayName: 'Test package creation'

      - script:
        inputs:
          workingDirectory: 'backend'
        steps:
          - task: RunShellScript@0
            inputs:
              script: |
                sudo apt-get update & sudo apt-get install -y zip
                sudo apt-get install -y curl
            displayName: 'Install curl and zip'

      - task: Archive@2
        inputs:
          rootFolderOrFile: 'backend'
          includeRootFolder: false
          archiveType: 'zip'
          archiveFile: '$(Build.ArtifactStagingDirectory)/backend.zip'
          replaceExistingArchive: true
        displayName: 'Archive backend'
```

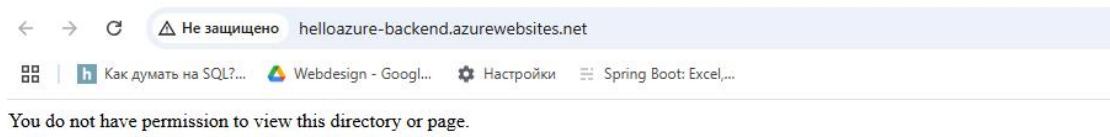
The screenshot shows the Azure DevOps interface for a project named "MyProjectVivi". The left sidebar lists various project sections: Overview, Boards, Repos, Files, Commits, Pushes, Branches, Tags, Pull requests, Advanced Security, Pipelines, Test Plans, and Artifacts. The "Files" section is currently selected. The main area displays the file structure of the "fullstack-azure-app" repository. Under the "backend" folder, there are ".gitignore", "index.js", and "index.test.js". Under the "frontend" folder, there are "package-lock.json", "package.json", and "index.html". Under the "infrastructure" folder, there are "azur...-backend.yml" and "azur...-frontend.yml". Other files listed include ".npmrc", "README.md", and "variables.yml". On the right side, the "index.js" file is open in a code editor. The code defines an Express app, sets up CORS, and handles a single endpoint that returns a hello message from the backend.

```
1 const express = require("express");
2 const cors = require("cors");
3 const app = express();
4
5 app.use(cors());
6
7 app.get("/", (req, res) => {
8   res.send("Hello Azure from Backend!");
9 });
10
11 module.exports = app;
12
13 if (require.main === module) {
14   const port = process.env.PORT || 3001;
15   const server = app.listen(PORT, () => {
16     console.log(`Server running on port ${PORT}`);
17   });
18   module.exports.server = server;
19 }
```

```

1 Starting: Deploy Backend to Azure
2 -----
3 Task : Azure Web App
4 Description : Deploy an Azure Web App For Linux or Windows
5 Version : 1.252.1
6 Author : Microsoft Corporation
7 Help : https://aka.ms/azurewebapptroubleshooting
8 -----
9 Got service connection details for Azure App Service:'helloazure-backend'
10 Updating App Service Application settings, Data: ["WEBSITE_RUN_FROM_PACKAGE":"1"] =>
11 Deleting App Service Application settings, Data: ["WEBSITE_RUN_FROM_ZIP"]
12 Updated App Service Application settings, and Kudu Application settings.
13 Package deployment using ZIP Deploy initiated.
14 Successfully deployed package to App Service.
15 NOTE: Run From Package makes website read-only, so you will receive an error when writing files to this directory.
16 Successfully updated App Service configuration details
17 App Service Application URL: http://helloazure-backend.azurewebsites.net
18 Finishing: Deploy Backend to Azure

```



Settings

Deploy and build code from your preferred source and build provider. Learn more

Your app is configured with Azure Pipelines as a build provider. Go to the [Azure DevOps portal](#) to manage your deployments.

Source	Azure Repos
Disconnect	

Azure Repos

Project	MyProjectVivi
Repository	fullstack-azure-app
Branch	refs/heads/main

Build

Build provider	Azure Pipelines
----------------	-----------------

Bicep/Terraform

bicep

← → ⌂ dev.azure.com/VolodymyrDibrova0326/MyProjectVivi/_git/fullstack-a

Как думать на SQL?... Webdesign - Googl... Настройки Spring Boot: Exc

main.bicep

Contents History Compare Blame

① You updated 89 AddingTest 8m ago

```
20 param sqlAdminLogin string = 'sqladmin'
21
22 @secure()
23 @description('Administrator password for SQL Server')
24 param sqlAdminPassword string = 'qwerty12345!'
25
26 // ----- App Service Plan -----
27 resource appServicePlan 'Microsoft.Web/serverfarms@2021-02-01' = {
28   name: appServicePlanName
29   location: location
30   sku: {
31     name: 'B1'
32     tier: 'Basic'
33   }
34 }
35
36 // ----- Frontend Web App -----
37 resource frontendApp 'Microsoft.Web/sites@2021-02-01' = {
38   name: frontendAppName
39   location: location
40   properties: {
41     serverFarmId: appServicePlan.id
42   }
43 }
44
45 // ----- Backend Web App -----
46 resource backendApp 'Microsoft.Web/sites@2021-02-01' = {
47   name: backendAppName
48   location: location
49   properties: {
50     serverFarmId: appServicePlan.id
51   }
52 }
53
54 // ----- SQL Server -----
55 resource sqlServer 'Microsoft.Sql/servers@2023-08-01' = {
56   name: sqlServerName
57   location: location
58   properties: {
59     administratorLogin: sqlAdminLogin
60     administratorLoginPassword: sqlAdminPassword
61   }
62 }
```

Terraform

main.tf

Contents History Compare Blame

ⓘ You updated 99 AddingTest 9m ago

```
5 resource "azurerm_resource_group" "helloazure" {
6   name     = "helloazure-rg"
7   location = "West Europe"
8 }
9
10 resource "azurerm_app_service_plan" "plan" {
11   name          = "helloazure-plan"
12   location      = azurerm_resource_group.helloazure.location
13   resource_group_name = azurerm_resource_group.helloazure.name
14   kind          = "linux"
15   reserved      = true
16
17   sku {
18     tier = "Basic"
19     size = "B1"
20   }
21 }
22
23 resource "azurerm_app_service" "frontend" {
24   name          = "helloazure-frontend"
25   location      = azurerm_resource_group.helloazure.location
26   resource_group_name = azurerm_resource_group.helloazure.name
27   app_service_plan_id = azurerm_app_service_plan.plan.id
28 }
29
30 resource "azurerm_app_service" "backend" {
31   name          = "helloazure-backend"
32   location      = azurerm_resource_group.helloazure.location
33   resource_group_name = azurerm_resource_group.helloazure.name
34   app_service_plan_id = azurerm_app_service_plan.plan.id
35 }
36
37 resource "azurerm_sql_server" "sqlserver" {
38   name           = "helloazure-sqlsrv"
39   resource_group_name = azurerm_resource_group.helloazure.name
40   location       = azurerm_resource_group.helloazure.location
41   version        = "12.0"
42   administrator_login = "sqladmin"
43   administrator_login_password = "My12345"
44   public_network_access = "Enabled"
45 }
```

Successfully deployed bicep

/ MyProjectVvii / Pipelines / fullstack-azure-app / 20250223.6

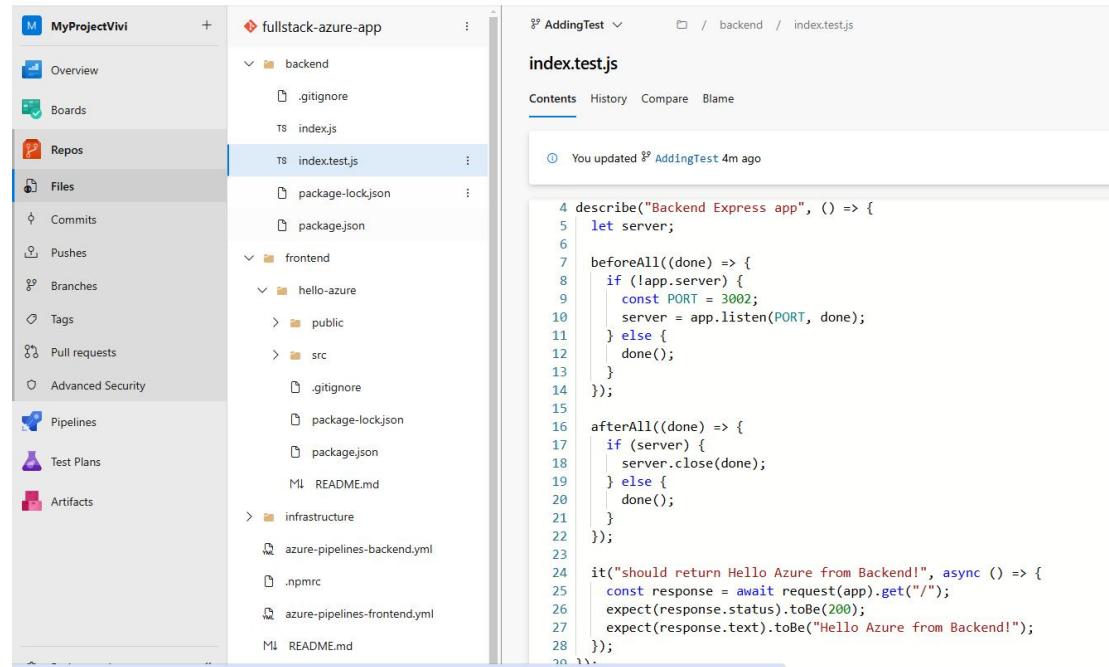
Jobs in run #20250223.6

Job	Status	Duration
Initialize job	Passed	<1s
Checkout fullstack-azure-app...	Passed	2s
Install Nodejs	Passed	<1s
Check if package.json exists ...	Passed	<1s
npm install (auth to Azure Art...	Passed	12s
Create test-results directory	Passed	<1s
Run Jest tests	Passed	1s
Publish Jest Test Results	Passed	3s
Build frontend	Passed	9s
Check build output	Passed	<1s
Copy .npmrc to frontend	Passed	<1s
Test package creation	Passed	1s
Publish to Azure Artifacts	Passed	3s
Install zip utility	Passed	2s
Archive frontend build	Passed	<1s
Install/Update Azure CLI on sel...	Passed	0s
Deploy Infrastructure wit...	In Progress	1m 14s
Deploy Frontend to Azure	Queued	26s
Post-job: Checkout fullstack-azur...	Queued	

Deploy Infrastructure with Bicep

```
7 Help      : https://docs.microsoft.com/azure/devops/pipelines/tasks/deploy/azure-cli
8 =====
9 /usr/bin/az --version
10 azure-cli      2.69.0
11
12 core          2.69.0
13 telemetry     1.1.0
14
15 Dependencies:
16 msal          1.31.2b1
17 azure-mgmt-resource 23.1.1
18
19 Python Location '/opt/az/bin/python3'
20 Config directory '/home/azureuser/.azure'
21 Extensions directory '/home/azureuser/.azure/clicextensions'
22
23 Python (Linux) 3.12.8 (main, Feb 5 2025, 06:39:23) [GCC 11.4.0]
24
25 Legal docs and information: aka.ms/AzureCliLegal
26
27
28 Your CLI is up-to-date.
29 Setting AZURE_CONFIG_DTR env variable to: /home/azureuser/magent/_work/_temp/_arclitask_@0
30 Setting active cloud to: AzureCloud
31 /usr/bin/az cloud set -n AzureCloud
32 /usr/bin/az login --service-principal -u *** --tenant 8d1157bb-1f96-415f-824b-ab0a29485d7d --allow-no-subscriptions --federated-token ***
33 [
34   (
35     "cloudName": "AzureCloud",
36     "homeTenantId": "8d1157bb-1f96-415f-824b-ab0a29485d7d",
37     "id": "98a6a428-dc3-44fe-bdf2-4e08593901a0",
38     "isDefault": true,
39     "managedByTenants": [],
40     "name": "Azure subscription 1",
41     "state": "Enabled",
42   )
43 ]
```

Test:

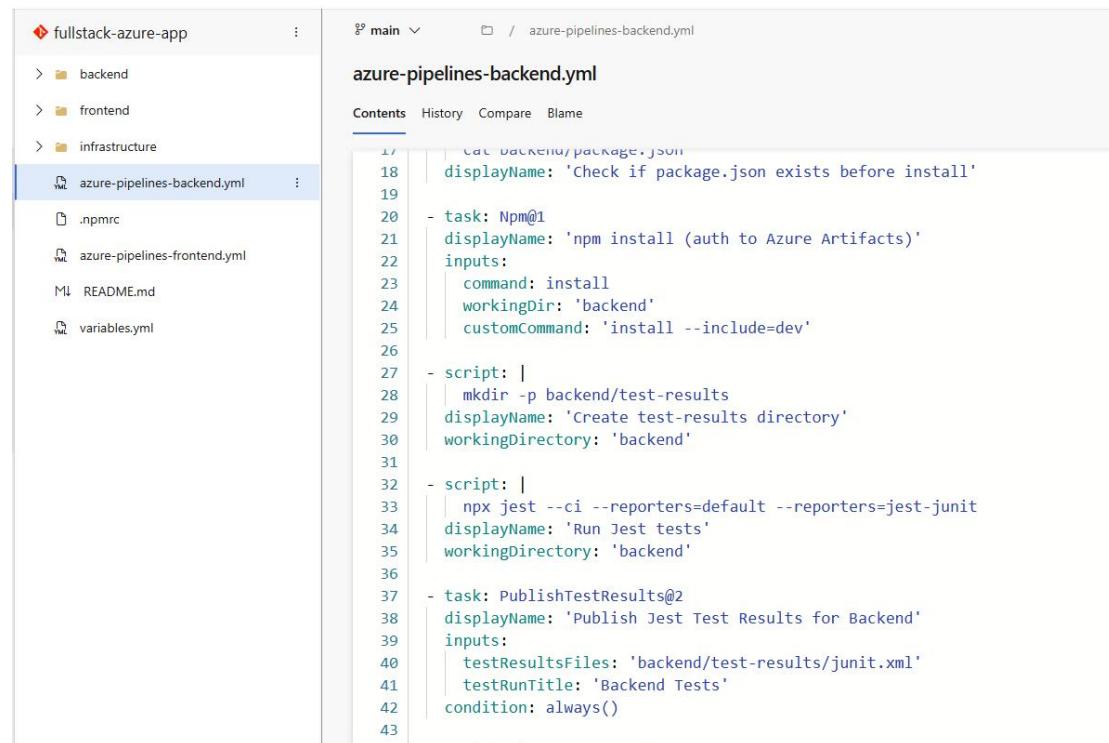


The screenshot shows the Azure DevOps interface for a repository named 'fullstack-azure-app'. The left sidebar lists various project sections like Overview, Boards, Repos, Files, Pipelines, Test Plans, and Artifacts. The 'Files' section is selected, showing a tree view of files. The 'index.test.js' file under the 'backend' directory is selected and shown in the main editor area. The code in the editor is a Jest test for an Express app, checking if it returns 'Hello Azure from Backend!'.

```

4 describe("Backend Express app", () => {
5   let server;
6
7   beforeAll((done) => {
8     if (lapp.server) {
9       const PORT = 3002;
10      server = app.listen(PORT, done);
11    } else {
12      done();
13    }
14  });
15
16  afterAll((done) => {
17    if (server) {
18      server.close(done);
19    } else {
20      done();
21    }
22  });
23
24  it("should return Hello Azure from Backend!", async () => {
25    const response = await request(app).get("/");
26    expect(response.status).toBe(200);
27    expect(response.text).toBe("Hello Azure from Backend!");
28  });
29});

```



The screenshot shows the Azure DevOps interface for the same repository. The left sidebar shows the 'azure-pipelines-backend.yml' file is selected. The main editor area displays the YAML configuration for the pipeline. It includes tasks for npm install, creating a test results directory, running Jest tests, and publishing test results.

```

17 | cat backend/package.json
18 | displayName: 'Check if package.json exists before install'
19 |
20 - task: Npm@1
21   displayName: 'npm install (auth to Azure Artifacts)'
22   inputs:
23     command: install
24     workingDir: 'backend'
25     customCommand: 'install --include=dev'
26
27 - script: |
28   | mkdir -p backend/test-results
29   displayName: 'Create test-results directory'
30   workingDirectory: 'backend'
31
32 - script: |
33   | npx jest --ci --reporters=default --reporters=jest-junit
34   displayName: 'Run Jest tests'
35   workingDirectory: 'backend'
36
37 - task: PublishTestResults@2
38   displayName: 'Publish Jest Test Results for Backend'
39   inputs:
40     testResultsFiles: 'backend/test-results/junit.xml'
41     testRunTitle: 'Backend Tests'
42     condition: always()
43

```

fullstack-azure-app

- > backend
- < frontend
 - > hello-azure
 - > public
 - > src
 - .gitignore
 - package-lock.json
 - package.json
 - M README.md
 - > infrastructure
 - azure-pipelines-backend.yml
 - .npmrc
 - azure-pipelines-frontend.yml
 - M README.md
 - variables.yml

azure-pipelines-frontend.yml

Contents History Compare Blame

You updated % AddingTest 3m ago

```

13 - task: NodeTool@0
14   inputs:
15   | versionSpec: '20.x'
16   | displayName: 'Install Node.js'
17
18 - script: |
19   | ls -la frontend/hello-azure
20   | cat frontend/hello-azure/package.json
21   | displayName: 'Check if package.json exists before install'
22
23 - task: Npm@1
24   | displayName: 'npm install (auth to Azure Artifacts)'
25   | inputs:
26   | | command: install
27   | | workingDir: 'frontend/hello-azure'
28
29 - script: |
30   | mkdir -p test-results
31   | displayName: 'Create test-results directory'
32   | workingDirectory: 'frontend/hello-azure'
33
34 - script: |
35   | npm test
36   | displayName: 'Run Jest tests'
37   | workingDirectory: 'frontend/hello-azure'
38
39 - task: PublishTestResults@2
40   | displayName: 'Publish Jest Test Results'
41   | inputs:
42   | | testResultsFiles: 'frontend/hello-azure/test-results/junit.xml'
43   | | testRunTitle: 'Jest Tests'
44   | | condition: always()
45

```

MyProjectVivi

Jobs

Jobs in run #20250223.21

	Job	Duration
Initialize job	<1s	
Checkout fullstack-azure-app...	2s	
Install Node.js	<1s	
Check if package.json exists ...	<1s	
npm install (auth to Azure Art...	11s	
Create test-results directory	<1s	
Run jest tests	1s	
Publish Jest Test Results	4s	
Build frontend	9s	
Check build output	<1s	
Copy .npmrc to frontend	<1s	

Publish Jest Test Results

```

1 Starting: Publish Jest Test Results
2 -----
3 Task : Publish Test Results
4 Description : Publish test results to Azure Pipelines
5 Version : 2.248.3
6 Author : Microsoft Corporation
7 Help : https://docs.microsoft.com/azure/devops/pipelines/tasks/test/publish-test-results
8 -----
9 Result Attachments will be stored in LogStore
10 Run Attachments will be stored in LogStore
11 Async Command Start: Publish test results
12 Publishing test results to test run '6'.
13 TestResults To Publish 1, Test run id:6
14 Test results publishing 1, remaining: 0. Test run id: 6
15 Published Test Run : https://dev.azure.com/VolodymyrDibrov0326/MyProjectVivi/_TestManagement/Buns?runId=6&runCharts
16 Async Command End: Publish test results
17 Finishing: Publish Test Test Results

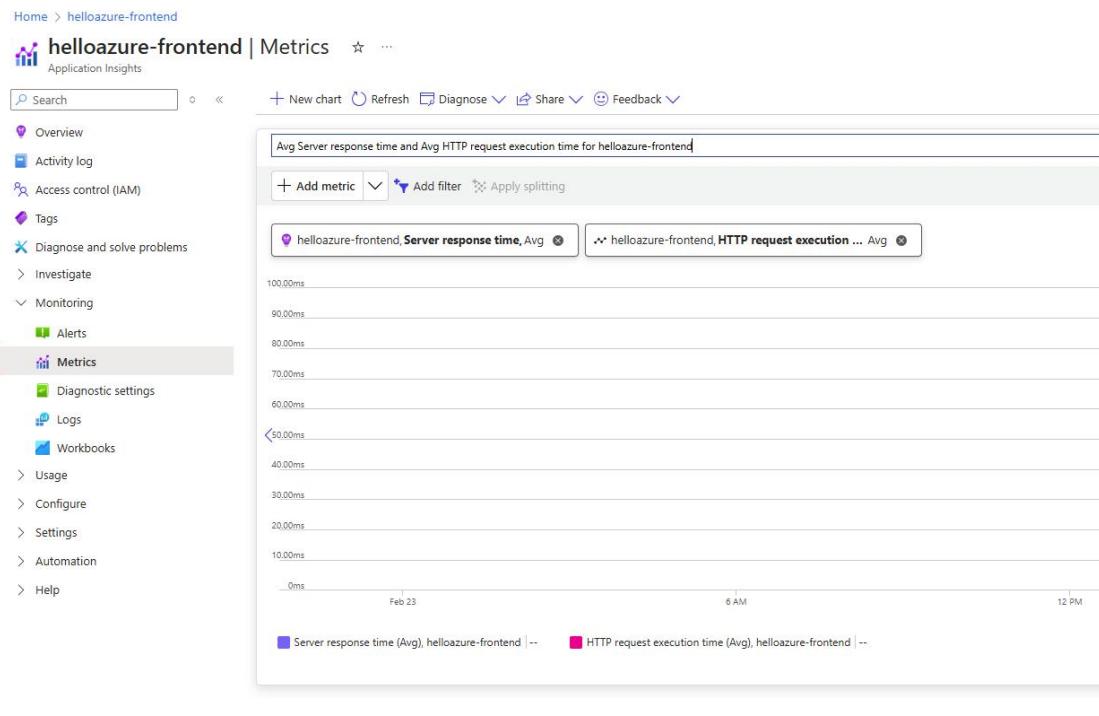
```

MyProjectVivi

Recent test runs

State	Run L.	Title	Completed Date	Build Number	Failed	Pass Rate
Completed	6	Jest Tests	2/23/2023 6:40:54 PM	20250223.21	0	100%
Completed	5	Jest Tests	2/23/2023 6:39:26 PM	20250223.20	0	100%
Completed	3	Jest Tests	2/21/2023 5:49:02 PM	20250221.12	0	100%
Completed	2	Jest Tests	2/21/2023 5:23:32 PM	20250221.4	1	0%

Monitoring without simulation



The screenshot shows the Azure Alert rules blade for the 'Azure subscription 1' scope. It lists two alert rules:

Name	Condition	Severity	Target scope	Target resource type	Signal type	Status
FailedRequest	requests/failed > 5	1 - Error	helloazure-frontend	Application Insights	Metrics	Enabled
Failure Anomalies - helloazure-frontend	Failure Anomalies detected	3 - Informational	helloazure-frontend	Application Insights	Smart detector	Enabled